Contents

1 INTRODUCTION

1.1 Project Profile

The face detection software is a windows form project developed in visual studio 2010. The project aims at making the image processing quick and simple as possible.The facial recognition has been a problem worked on around the world for many persons; this problem has emerged in multiple fields and sciences, especially in computer science, others fields that are very interested In this technology are: Mechatronic, Robotic, criminalistics, etc. I work in this interesting topic using EmguCV cross platform .Net wrapper to the Intel OpenCV image processing library and C # .Net, these librarys allow me capture and process image of a capture device in real time. The main goal of this project is show and explains the easiest way how implement a face detector and recognizer in real time for multiple persons using Principal Component Analysis (PCA) with eigenface for implement it in multiple fields.It can access only the admin.

Facial recognition is a computer application composes for complex algorithms that use mathematical and matricial techniques, these get the image in raster mode(digital format) and then process and compare pixel by pixel using different methods for obtain a faster and reliable results, obviously these results depend of the machine use to process this due to the huge computational power that these algorithms, functions and routines requires, these are the most popular techniques used for solve this modern problem.

1.2 Scope of The Project

The Scope of this project is to develop software for marking student attendance digitaly. This software is developed in the programming platform of ASp.net and MySQL for managing database. The input screen has developed with care to make it user friendly and avoids errors at the time of output. Documentation is prepared by mean of activity diagram and flowchart. They have been developed in such a way that it is very much self-explanatory and clear.

The face recognition attendance system, is concerned with the development and maintenance of mutually beneficial . Its focus is the creation of long-term value, and not just short-term profits, for the company and all it works with. The scope of face recognition is higher in the future because all operations are done computerized.Compairing to the past all activites are done by manually,It waste the time very huge.So this system can provide more scope in the world

2 ABOUT THE DEVELOPING TOOLS

2.1 Introduction to visual studio

Microsoft visual studio is the main integrated development environment (IDE) from Microsoft. It can be used to console and graphical user interface applications along with windows forms applications, websites, web applications, and web services in both native code together with managed code for all platform supported by Microsoft Windows, Windows Mobile.Net Framework and .Net Compact Framework.

Visual studio supports languages by means language services, which allow any programming language to be supported by the code editor and debugger, provided a language-specific service has been authored. Built in language include C/C++, VB.NET, (via Visual Basic .NET), and C#(via visual C#).It also supports XML, HTML, java Script and CSS. Language-specific version of visual studio also exist which provide more limited language services to the user .These individual package are called Microsoft visual Basic ,visual J#, visual C#, visual C++. C#.NET is a programming framework built on the common language run time that can be used on a server to build powerful system based applications.

2.2 Introduction to ASP.Net

ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites. It allows you to use a full featured programming language such as C# or VB.NET to build web applications easily. ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices. ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation. ASP.NET is a part of Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework.The ASP.NET application codes can be written in any of the following languages:

1. C#

2.Visual Basic.Net

3. Jscript

4. J #

ASP.NET is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls such as text boxes, buttons, and labels for assembling,configuring and manipulating code to create HTML pages.ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web develop- ment models:

1. Enhanced Performance: ASP.NET is compiled common language runtime code running on the server. Unlike interpreted predecessors, ASP.NET cantake advantage of early binding, just-in-time compilation, native optimization and caching services right out of the box. This amount is dramatically better performance before you ever write a line of code.

2. World-class Tool Support: A rich toolbox and designer in the visual studio integrated development environment complement of the ASP.NET framework. WYSIWYG editing, drag-and-drop server controls and automatic deployment are just a few of the features this powerful tool provides.

3. Power and Flexibility: Because ASP.NET is based on the common language runtime, the power and exibility of that entire platform is available to Web application developers. The .NET framework class library, Messaging and Data Access solutions are all seamlessly accessible from the Web.

4. Customizability and Extensibility: ASP.NET delivers a well-factored architecture that allows developers to plug-in their code at appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET with your own custom-written component. Implementing custom authentication or state services has never been easier.

5. Scalability and Availability: ASP.NET has been designed with scalability in mind, with features specificaly tailored to improved performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET Runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps your application constantly available to handle requests.

6. Simplicity:-C#.NET makes it easy to perform common task, from simple from submission and client authentication to deployment and site configuration .For example, the C#.NET from the framework allows you to build user interface that cleanly separate application logic from presentation code and to handle events in a simple, visual Basic -like forms processing model. Additionally the common language run time simplifies development, with managed code service such as automatic reference counting and garbage collection.

7. Manageability:-C#.NET employs a text-based, hierarchical configuration system, which simplifies applying setting to your server environment and web application .Because configuration information is stored as plain text, new setting may be applied without the aid of local administration tools .This zero local administration philosophy extends to deploying C#.NET Framework application as well.

8. Security:-With built in Windows authentication and per-application configuration, you can be assured that your application are secure.

9. Language Support:-The Microsoft .NET platform currently offers built-in support for three languages: C#, visual Basic, and J#. The exercises and code sample in this tutorial demonstrate how to use C#, Visual Basic, and J# to build.NET applications

2.2.1 C # and Features

Microsoft C# is a new programming language designed for building a wide range of enterprise applications that run on the .NET Framework. C# code is compiled as managed code, which means it benifits from the services of the common language runtime. These services include language interoperability, garbage collection, enhanced security and improved versioning support. The C# language is an evolution of C and C-H-. It uses many C++ features in the areas of statements, expressions and operators. C# provides access to the common API styles: .NET Framework, COM, Automation and C-style APIs. It also supports unsafe mode, where you can use pointers to manipulate memory that is not under the control of the garbage collector.

C# is introduced as Visual C# in the Visual Studio, Net suit. Support for Visual C# includes project templates, designers, property pages, code wizards, an object model and other features of the development environment. The library for Visual C# programming is the .NET Framework. C# is an elegant, simple, type-safe, object-oriented language that allows enterprise programmers to build a breadth of applications. C# also gives you the capability

1. Full COM/Platform support for existing code integration.

2. Robustness through garbage collection and type safety.

3. Security provided through intrinsic code trust mechanisms.

4. Full support of extensible metadata concepts.

2.3 MySQL

MySQL (officially pronounced as "My S-Q-L",) is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.For proprietary use, several paid editions are available, and offer additional functionality. MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python"

SQL Server is the native data store of C# .NET every business enterprise maintains large volumes of data for its operations. With more and more people accessing data for their work, the need to maintain its integrity and relevance increases. Normally with the traditional method of storing data and information in the files, the chances that the data loses, its integrity and validity are very high. SQL Server 2008 offers capabilities of both relational and object oriented database systems. In general, objects can be defined as reusable software calls which can be location independent and perform a specific task on any application environment with little or no chance to the code.

SQL Server products are based on a concept known as Client Server Technology. This concept invokes segregating the processing of an application between two systems. One performs all activities related to database (server) and other performs activities that help the user to interact with the application client. The MySQL software delivers a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. MySQL is a trademark of Oracle Corporation and/or its affiliates, and shall not be used by Customer without Oracle's express written authorization. Other names may be trademarks of their respective owners.

2.3.1 Features of MySQL

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server.MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

1. Cross-platform support
2. Stored procedures, using a procedural language that closely adheres to SQL/PSM
3. Triggers
4. Updatable views
5. Information schema
6. Built-in replication support (i.e., master-master replication and master-slave replication) with one master per slave, many slaves per master.
7. SSL support
8. Query caching

2.3.2 Limitations of MySQL

1. When using some storage engines other than the default of InnoDB, MySQL does not comply with the full SQL standard for some of the implemented functionality, including foreign key references and check constraints.
2. Up until MySQL 5.7, triggers are limited to one per action / timing, meaning that at most one trigger can be defined to be executed after an INSERT operation, and one before INSERT on the same table.[80] No triggers can be defined on views.

2.3.3 User Interfaces

A graphical user interface (GUI) is a type of interface that allows users to interact with electronic devices or programs through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation. GUIs are easier to learn than command-line interfaces (CLIs), which require commands to be typed on the keyboard. Third-party proprietary and free graphical administration applications (or "front ends") are available that integrate with MySQL and enable users to work with database structure and data visually.

2.4 GIT

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development,but it can be used to keep track of changes in any set of files. As a distributed revision control system it is aimed at speed, data integrity,and support for distributed, non-linear workflows.Git was created by Linus Torvalds in 2005 for development of the Linux kernel, with other kernel developers contributing to its initial development.[12] Its current maintainer since 2005 is Junio Hamano.As with most other distributed version control systems, and unlike most clientserver systems, every Git directory on every computer is a full-fledged repository with complete history and full version tracking abilities, independent of network access or a central server.Git is free software distributed under the terms of the GNU General Public License version 2.

2.4.1 Characteristics

Git's design is a synthesis of Torvalds's experience with Linux in maintaining a large distributed development project, along with his intimate knowledge of file system performance gained from the same project and the urgent need to produce a working system in short order. These influences led to the following implementation choices;

1. Strong support for non-linear development
2. Distributed development
3. Compatibility with extant systems and protocols
4. Efficient handling of large projects
5. Cryptographic authentication of history
6. Garbage accumulates until collected
7. Periodic explicit object packing

Another property of Git is that it snapshots directory trees of files. The earliest systems for tracking versions of source code, Source Code Control System (SCCS) and Revision Control System (RCS), worked on individual files and emphasized the space savings to be gained from interleaved deltas

(SCCS) or delta encoding (RCS) the (mostly similar) versions. Later revision control systems maintained this notion of a file having an identity across multiple revisions of a project. However, Torvalds rejected this concept.Consequently, Git does not explicitly record file revision relationships at any level below the source code tree.Git implements several merging strategies; a non-default can be selected at merge time:

1. resolve: the traditional three-way merge algorithm.
2. recursive: This is the default when pulling or merging one branch, and is a variant of the three-way merge algorithm.
3. octopus: This is the default when merging more than two heads.

2.4.2 Implementations

Git is primarily developed on Linux, although it also supports most major operating systems including BSD, Solaris, macOS, and Windows.The first Microsoft Windows port of Git was primarily a Linux emulation framework that hosts the Linux version. Installing Git under Windows creates a similarly named Program Files directory containing the MinGW port of the GNU Compiler Collection, Perl 5, msys2.0 (itself a fork of Cygwin, a Unix-like emulation environment for Windows) and various other Windows ports or emulations of Linux utilities and libraries. Currently native Windows builds of Git are distributed as 32 and 64-bit installer.The JGit implementation of Git is a pure Java software library, designed to be embedded in any Java application. JGit is used in the Gerrit code review tool and in EGit, a Git client for the Eclipse IDE.The Dulwich implementation of Git is a pure Python software component for Python 2.7, 3.4 and 3.5.The libgit2 implementation of Git is an ANSI C software library with no other dependencies, which can be built on multiple platforms including Windows, Linux, macOS, and BSD.It has bindings for many programming languages, including Ruby, Python, and Haskell.JS-Git is a JavaScript implementation of a subset of Git.

2.4.3 GIT Server

As Git is a distributed version control system, it can be used as a server out of the box. Dedicated Git server software helps, amongst other features, to add access control, display the contents of a Git repository via the web, and help managing multiple repositories. Remote file store and shell access: A Git repository can be cloned to a shared file system, and accessed by other persons. It can also be accessed via remote shell just by having the Git software installed and allowing a user to log in.

2.4.4 Security

Git does not provide access control mechanisms, but was designed for operation with other tools that specialize in access control.An attacker could perform arbitrary code execution on a target computer with Git installed by creating a malicious Git tree (directory) named .git (a directory in Git repositories that stores all the data of the repository) in a different case (such as .GIT or .Git, needed because Git doesn't allow the all-lowercase version of .git to be created manually) with malicious files in the .git/hooks subdirectory (a folder with executable files that Git runs) on a repository that the attacker made or on a repository that the attacker can modify.

If a Windows or Mac user pulls (downloads) a version of the repository with the malicious directory, then switches to that directory, the .git directory will be overwritten (due to the case-insensitive trait of the Windows and Mac filesystems) and the malicious executable files in .git/hooks may be run, which results in the attacker's commands being executed. An attacker could also modify the .git/config configuration file, which allows the attacker to create malicious Git aliases (aliases for Git commands or external commands) or modify extant aliases to execute malicious commands when run. The vulnerability was patched in version 2.2.1 of Git, released on 17 December 2014, and announced on the next day.

2.5 GitHub

GitHub (originally known as Logical Awesome LLC) is a web-based hosting service for version control using git. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

GitHub offers plans for both private repositories and free accounts which are commonly used to host open-source software projects.As of April 2017, GitHub reports having almost 20 million users and 57 million repositories, making it the largest host of source code in the world.

2.5.1 Services

Development of the GitHub platform began on October 19, 2007. The site was launched in April 2008 by Tom Preston-Werner, Chris Wanstrath, and PJ Hyett after it had been made available for a few months prior as a beta release. Projects on GitHub can be accessed and manipulated using the standard Git command-line interface and all of the standard Git commands work with it. GitHub also allows registered and non-registered users to browse public repositories on the site. Multiple desktop clients and Git plugins have also been created by GitHub and other third parties that integrate with the platform.

The site provides social networking-like functions such as feeds, followers, wikis (using wiki software called Gollum) and a social network graph to display how developers work on their versions (”forks”) of a repository and what fork (and branch within that fork) is newest. A user must create an account in order to contribute content to the site, but public repositories can be browsed and downloaded by anyone. With a registered user account, users are able to discuss, manage repositories, submit contributions to others' repositories, and review changes to code. The software that runs GitHub was written using Ruby on Rails and Erlang by GitHub, Inc. developers Chris Wanstrath, PJ Hyett, and Tom Preston-Werner.

3 SYSTEM ANALYSIS

3.1 Introduction

System Analysis is a structured method for solving the problems related to the development of a new system. The detailed investigation of the present system is the focal point of system analysis. The main aim of the system is to provide the organization with efficient and user-friendly automation. So the system analysis process should be performed with extreme precision so that an accurate picture of the existing system, disadvantages and the requirements of the new system can be obtained.

System analysis is a general term that refers to an orderly, structured process for identifying and solving problems. We call system analysis process life cycle methodology, since it relates to four significant phases .They are:

1. Study phase
2. Design phase
3. Development phase
4. Implementation phase

Analysis implies the process of breaking something into parts so that the whole maybe understood. The definition of system analysis includes not only the process of analysis but also that of synthesis, which implies the process of putting together to form a new one. All activities associated with each life cycle phase must be performance, management,documentation of the activities related to the life cycle phases of a computer based business system. In the study phase a detailed study of the project is made and clear picture of the project should be in mind by this time. In the design phasethe designing of the input, output and table designs are made. In the development phase is where the physical designing of the input- output screens and coding of the system is done. In the system implementation actually implements the system by making necessary testing.

3.2 Existing System

A newly emerging trend, claimed to achieve previously unseen accuracies, is three-dimensional face recognition. This technique uses 3-D sensors to capture information about the shape of a face. This information is then used to identify distinctive features on the surface of a face, such as the contour of the eye sockets, nose, and chin. One advantage of 3-D facial recognition is that it is not affected by changes in lighting like other techniques. It can also identify a face from a range of viewing angles, including a profile view. Even a perfect 3D matching technique could be sensitive to expressions. For that goal a group at the Technion applied tools from metric geometry to treat expressions as isometries.

Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. This technique, called skin texture analysis, turns the unique lines, patterns, and spots apparent in a persons skin into a mathematical space Tests have shown that with the addition of skin texture analysis, performance in recognizing faces can increase 20 to 25 percent. It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems.

3.2.1 Disadvantage of existing system

1. 3D facial recognition needs a 3d camera to operate.
2. Only work on ultra light sensitive camera..
3. It requires more time.
4. Cost is high

3.3 Feasibility study

The prime objective of feasibility study is to ensure that the problem is worth to be solved. At this stage a cost benefit analysis is performed to assertion that the benefit from the system will over rule the cost associated with the whole analysis, design and development of the new system. An important outcome of the preliminary investigation determining whether the system required is feasible.

Feasibility study is a test of proposed system regarding its efficiency, impact on the organization, ability to meet the needs of users and effective use of resources. Thus, when a new project is proposed, it normally goes through a feasibility study before it is approved for development.

When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration. During feasibility analysis for this project, following primary areas of interest facts considered in the feasibility analysis were:

1. Technical Feasibility
2. Economic Feasibility
3. Operational Feasibility
4. Social Feasibility
5. Behavioural Feasibility

3.3.1 Technical Feasibility

It is a study of resources availability that may affect the availability to achieve an acceptable system. It is essential that the process of analysis and definition to conducted in parallel with the assessment of technical feasibility. It centres on the existing computer system and to what extent it can support to the proposed system. This involves the financial considerations to accommodate technical enhancements .If the budgets is a serious of constraint, the project is judged as not feasible. The handling of the proposed system does not Require the changing of the existing configuration of the system.

The technical needs of the system may include: frontend and backend selection. An important issue for the development of a project is the selection of suitable front end and backend. When we decided to develop the project we went through an extensive study to determine the most suitable platform that suits the needs of the organisation as well as helps in development of the project. The aspect of study included several factors. Based on those factors we selected PHP as the frontend and MySQL as the backend for developing our project. This project is technically feasible as required software is easily available.

3.3.2 Economic Feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of the system and is commonly known as cost benefit analysis, the procedure made costs. The result of a comparison is found out and changed if needed. This is an on-going effort that improves the accuracy at each phase of the system life cycle. If a benefit outweighs costs, then decision is made to decide and implement the system. Otherwise, further justification or alternation in the proposed system will have to be made and the process is repeated. It has been proven that the proposed system is economically feasible since it provides several cost benefits. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system.

This feasibility study present tangible and intangible benefits from the project by comparing the development and operational cost. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This system needs some more initial investment than the existing system, but it can be justifiable that it will improve quality of service.

Thus feasibility study should centre along the following points:

1. Improvement resulting over the existing method in terms of accuracy, timeliness
2. Cost comparison
3. Estimate on the life expectancy of the hardware
4. Overall objective.

Our project is economically feasible. It does not require much cost to be involved in the overall process. The overall objective is in easing out the recruitment processes.

3.3.3 Operational Feasibility

Operational feasibility is looked at in if the propose solution fitting with current operations. The proposed project would be beneficial to fortune, that it satisfies the objectives when developed and installed. One of the main problems faced during development of a new system is getting acceptance from the user. There is support from the management of fortune, towards the development of the project. All the operational aspects are considered carefully. Thus the project is operationally feasible.

The system was found to be technically, economically and operationally feasible. The system developed is user friendly, needs less training and improve the working environment.

3.3.4 Social Feasibility

This is concerned with the effect on employees and customers on the introduction of a new system. Will it result in redundancies? Will some jobs be deskilled? Is there a need for re-training? Will the workforce be able to cope with the new changes? Will the workforce has to relocate? It is imperative that users are being involved and their cooperation is secure before changes are made. Equally the effects on customer services has to be identified.

3.3.5 Behavioural Feasibility

This is also known as operational feasibility. Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. One of the main problems faced during the development of the new system is getting the acceptance from the user. People are inherently resistant to change and so estimate should be made of how strong a reaction the user is likely to have towards the developing system. The system is much user friendly and the maintenance and working needs much less human effort. Define the urgency of the problem and the acceptability of any solution; if the system is developed, will it be used? It includes people oriented and social issues: internal issues, such as manpower problems, labourobjections, manager resistance, organisational conflicts and policies; also external issues, including social acceptability, legalaspects and government regulations.

In case of this system, the organisation is completely in favour of creating the raid based shopping Management system as it saved their precious time, energy and moreover the system when implemented would help to remove inconsistencies, reduce manpower etc. Also there is no specialized training needed, only a few hours of instructed demo needs to be given to the user. So it might hence the system is behavioural feasible. This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and therefore it will accept broad audience from around the world.

3.4 Proposed system

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system in a cost effective way. The system is very fast as compared to the predecessors. System can optimized for specified hardware before it is supplied. The new system entitled 'face detection attendance system provides a lot of services to the user as compared to the existing system. It provides a better way for the users to get all the information needed as quick as possible . The user can get the information regarding different available service details. Admin can also view the attendance details of the students. Whenever a new face is added, he/she can add present in the next class. This site contains the administration module, which deals with modifications and updating of data.

Advantages of Proposed System:

1. Even with the vga camera the software works fine.
2. simple but very effective user interference.
3. Implementation requires very less components as considering with other technologies, thus this is a cheap and efficient.
4. Can optimized for better performance. If the computer system has excellent hardware configuration.
5. High speed
6. Transparency
7. Time saving
8. System dependent
9. Error free
10. Secure

4 FACT FINDING TECHNIQUES

The success of any project depends upon the accuracy of available data. Accurate information can be collected with the help of certain methods / techniques. These specific methods for finding information of the system are termed as fact finding techniques. Interview, Questionnaire, Record View and Observations are the different fact finding techniques used in this project.

4.1 Interview:

This method is used to collect the information from groups or individuals. We select the people who are related with the system for the interview. In this method, we sit face to face with people and record their responses.

4.2 Record View:

The information related to the system is available in the source like companys documents, websites and other records. This record review helped me to get valuable information about the system.

4.3 Onsite observation:

Unlike the other fact finding techniques, in this method we visit the organization and observe and understand the working of the existing system, flow of the system, the users of the system etc.

5 SYSTEM SPECIFICATION

5.1 Hardware Specification:-

The selection of hardware configuration is very important task related to software development. The processor should be powerful to handle all the operations. The hard disk should have the sufficient capacity to solve the database and the application.

The hardware requirements for developing and implementing the proposed system are given below:

1. Processor : Intel(R) pentium
2. RAM : 4.00GB
3. Hard Disk Drive : 500GB
4. Key Board : Standard 101/102 or Digi Sync Family
5. Monitor : Display Panel (1024 X 764)
6. Pointing device : mouse
7. Display Adapter : Trident Super VGA
8. Mouse : Logitech Serial Mouse

5.2 Software Specification:-

Windows XP server includes improved network, application, and Web services. It provides improved reliability and scalability, lowers yours cost of computing with powerful, flexible management services, and provides the best foundation for running business applications. It provides network data security by protecting data on the wire or at the network interface. It also provides stored data on the security by using data encryption. Data encryption is provided transparently within windows XP by feature known as Encrypting File System (EFS). It has the ability to run on a single PC chip with a user up to a multi-user, multi-processor network installation.

The software requirements for developing and implementing the proposed system are given below:

1. Operating System : Windows XP Professional or Higher
2. Front End : Asp.net.
3. Scripting Language : HTML
4. Back-end : MYSQL SERVER
5. IDE : Microsoft Visual Studio 2008

6 SYSTEM DESIGN

6.1 Introduction to System Design

The most creative and challenging phase of the system life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementing a candidate system. It also includes the construction of programs and program testing. The question here is: How should the problem is solved?

The first step is to determine how the output is to be produced and in what format. Samples of the output (and input) are also presented. Second input data and master files have to be designed to meet the requirement of the proposed output. The operational phase are handled through program construction and testing, including a list of the programs needed to meet the systems objectives and complete documentation. Finally, details related to justification of the system and an estimate of the impact of the candidate system on the user and the organization are documented and evaluated by management as a step toward implementation. The goal of design process is to produce a model as representation of a system, which can be used later to build that system. The produced model is called the design of the system. The design process for software systems often has two levels. At the first level, the focus is on deciding which modules are needed for the system, the specification of these modules, and how the modules should be interconnected. This is what is called the system design or top level design. In the second level, the internal design of the module can be satisfied, is decided. This design level is often called detailed design of logic design.

6.2 Input Design

Input design is the process of converting the user orientedinput into a computer-based system format. The design is a part of overall system design that needs careful attention.

The collection of input data is considered to be the most expensive part of the system design.It also includes determining the record media, method of input, speed of capture and entry on to the screen. Online data entry accepts commands and data through a keyboard. The major approach to input design is the menu and the prompt design. In each alternative, the users options are predefined. The data flow diagram indicates logical data flow, data stores, source and destination. Input data are collected and organized into a group of similar data once identified input media are selected for processing.

In this software, importance is given to develop Graphical User Interface (GUI), which is an important factor in developing efficient and user friendly software. For inputting user data, attractive forms are designed. User can also select the desired options from the menu, which provides all possible facilities. Also the important input format is designed in such a way that accidental errors are avoided. The user has to input only just the minimum data required, which also helps in avoiding the errors that the users may make accurate.

Designing of the input format is very important in developing efficient software. The goal of input design is to make entry as easy, logical and free from errors.The following are the objectives of input design:

1. To produce a cost effective method of input.
2. To make the input forms understandable to the end users.
3. To ensure the validation of data inputs.

The nature of input data is determined partially during logical system design. However the nature of inputs on the system is also determined. Efforts has been made to ensure that input data remains accurate from the stage at which it is recorded And documented to the stage at which it is accepted by the computer. Validation procedures are also present to detect errors in data input, which is beyond control procedures. Validation procedures are designed to check each record, data item or field against certain criteria.

6.3 Output Design

One of the most important features of an information system for users is the output it produces. Output is the information delivered to users through the information delivered to users through the information system. Users generally merit the system solely by its output. Hence the system analysts work closely with user through an interactive process, until the result is considered to be satisfactory.

The output design is an ongoing activity almost from the beginning of the project, and follows the principles of form design. Computer is the most important improves the relationship of the system and the user, thus facilitating decision-making. The primary considerations in the design of the output are the requirement of the information and the objectives of the end users. A major form of output is a hard copy from the printer, however soft copies are also available.

A major form of output is a hard copy obtained from the printer. These printouts are designed to include the exact requirements of the user. The outputs required by the end user are defined during the logical design stage in terms of:

1. The type of output
2. The contents
3. The format
4. The actions required

The contents of the output are then defined in detail during the physical design of outputs.
Objective of output design are:

1. Design output to serve the intended purposes
2. Design output to fit the user
3. Deliver the appropriate quantity of output
4. Assume that output is where it is needed
5. Provide output on time
6. Choose the right output method

6.4 DataBase Design

Database design is the process of producing a detailed data model of a database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS).

The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

1. Determine the data to be stored in the database.
2. Determine the relationships between the different data elements.
3. Superimpose a logical structure upon the data on the basis of these relationships.

6.5 Normalization

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

The database community has developed a series of guidelines for ensuring that databases are normalized. These are referred to as normal forms and are numbered from one (the lowest form of normalization, referred to as first normal form or 1NF) through five (fifth normal form or 5NF). In practical applications, you'll often see 1NF, 2NF, and 3NF along with the occasional 4NF. Fifth normal form is very rarely seen.

6.5.1 First Normal Form(1NF)

First normal form (1NF) sets the very basic rules for an organized database.Create separate tables for each group of related data and identify each row with the primary key.

6.5.2 Second Normal Form(2NF)

Second normal form (2NF) further addresses the concept of removing duplicative data.

6.5.3 Third Normal Form(3NF)

Third normal form (3NF) goes one large step further .Meet all the requirements of the second normal form and remove columns that are not dependent upon the primary key.

6.5.4 Boyce-Codd Normal Form (BCNF or 3.5NF)

The Boyce-Codd Normal Form, also referred to as the "third and half (3.5) normal form.

6.5.5 Fourth Normal Form(4NF)

Finally, fourth normal form (4NF) has one additional requirement.A relation is in 4NF if it has no multi-valued dependencies.

6.6 System Modules

The system is divided into 3 main processes:

1.Face detection

In the Face detection process. Multipple faces are identified in real time.Using a ordinary web camera.The opencv library help us to grab the frames from the cam buffer. And allow us to serch for a specific pattern face.The EmguCV cross platform .Net wrapper to the Intel OpenCV image processing library and Csharp .Net help us to do this task

2. Facial recognition

Facial recognition is a computer application composes for complex algorithms that use mathematical and matricial techniques, these get the image in raster mode(digital format) and then process and compare pixel by pixel using different methods for obtain a faster and reliable results, obviously these results depend of the machine use to process this due to the huge computational power that these algorithms, functions and routines requires

3.Compare images

In the compare images process. Two images are compared using Csharp.net image processing library. The library procedure decode the image. Extract the binary data of the image compare it bit by bit. Then archive the binary data. Encode it into image format

6.7 Table Structure

Table No:1

Table Name: Login

Primary Key: id

Field Name	**Type**	**Description**
id	int	
uname	varchar	username
password	varchar	password

Table No:2

Table Name: Attendance

Primary Key: roll no

Field Name	Type	Description
rollno	int	
name	varchar	Name of student
dept	varchar	department
date	date	Current date

6.8 UML

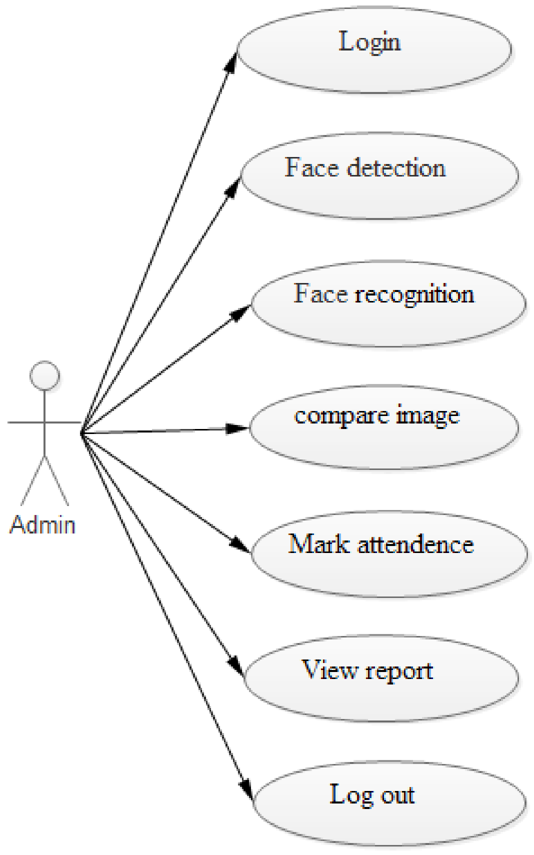


Figure 1: Use case

6.8.1 Activity Diagram

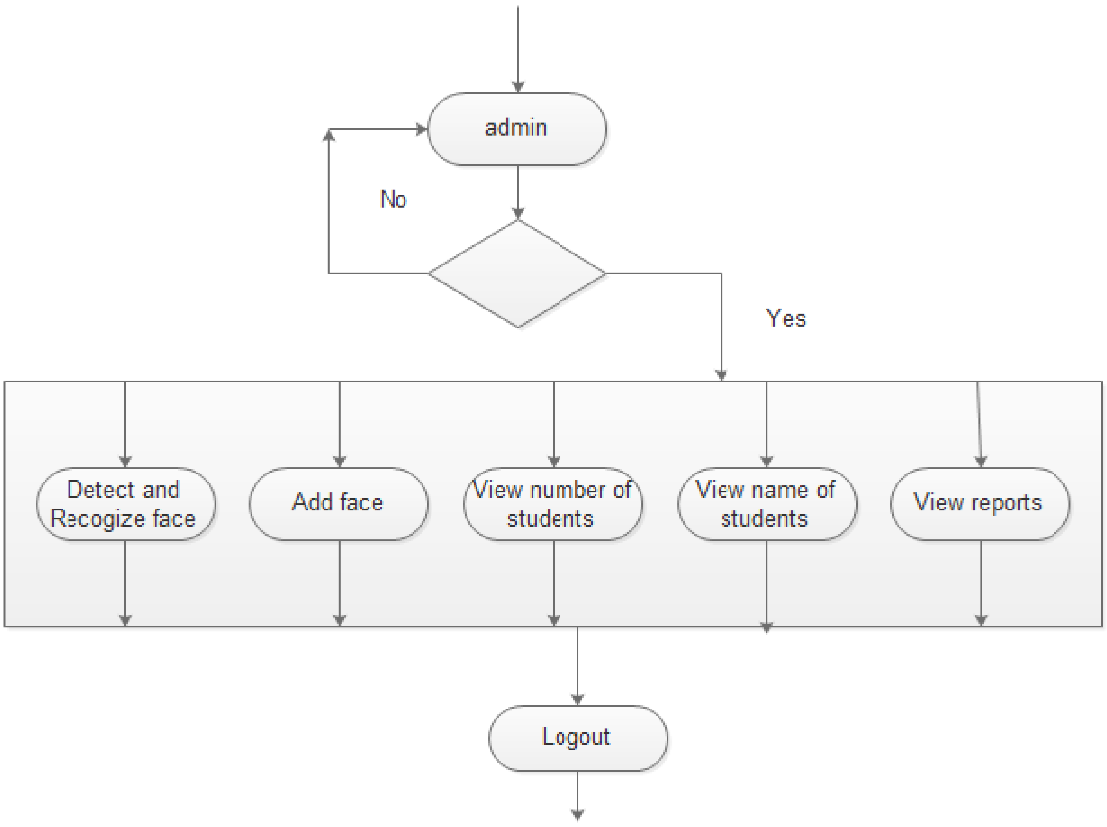


Figure 2: Activity-Admin

7 SYSTEM TESTING

7.1 Introduction to System Testing

Testing is an activity to verify that a correct system is being built and is performed with the intent of finding fault in the system.However not restricted to being performed after the development phase is complete. But this is to carry out in parallel with all stages of system development, starting with requirement specification. Testing results, once gathered and evaluated, provide a qualitative indication of software quality and reliability and serve as a basis for design modification if required. A project is said to be incomplete without proper testing.System testing can be broadly classified into:

1. Syntax testing
2. Unit testing
3. Integration testing
4. Validation testing
5. Output testing
6. User acceptance testing

Testing is the process of examining the software to compare the actual behavior with that of the excepted behavior. The major goal of software testing is to demonstrate that faults are not present. In order to achieve this goal the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present.

System testing is the first stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved.

7.2 Syntax Testing

System testing involves unit testing, integration testing, acceptance testing. Careful planning and scheduling are required to ensure that modules will be available for integration into the evolving software product when needed. A test plan has the following steps:

1. Prepare test plan
2. Specify conditions for user acceptance testing
3. Prepare test data for program testing
4. Prepare test data for transaction path testing
5. Plan user training
6. Compile/assemble programs
7. Prepare job performance aids
8. Prepare operational documents

7.3 Unit Testing

In computer programming, unit testing is a procedure used to validate that individual units of source code are working properly. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or derived/child class.Ideally, each test case is independent from the others; mock or fake objects as well as test harnesses can be used to assist testing a module in isolation. Unit testing is typically done by software developers to ensure that the code they have written meets software requirements and behaves as the developer intended.

In this we test each module individually but not integrate the whole system. It focuses verification efforts even in the smallest unit of software design in each module. This is also known as Module Testing. The testing is carried out in the programming style itself. In this testing each module is focused to work satisfactorily as regard to the expected output from the module. There are some validation checks for the fields.

7.4 Integration Testing

Integration testing (sometimes called Integration and Testing, abbreviated I and T) is the phase of software testing in which individual software modules are combined and tested as a group. It follows unit testing and precedes system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.Data can be lost across an interface, one module can have adverse effect on the other sub-functions, when combined may not produce the desired functions. Integration testing is the systematic testing to uncover the errors within the interface. This testing is done with simple data .The need for an integrated system is to find the overall performance.

The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing.

7.5 Validation Testing

At the culmination of black box testing (Here the structure of the program is not considered), software is completely assembled as a package .Interface errors have been uncovered and correct and final series of tests, i.e., and validation test begins. The customer defines validation with a simple definition and validation succeeds When the software functions in manner than can be reasonably accepted.

7.6 Output Testing

The output of the software should be acceptable to the system user. The output requirement is defined during the system analysis. Testing of the software system is done against the output and the output testing was completed with success.

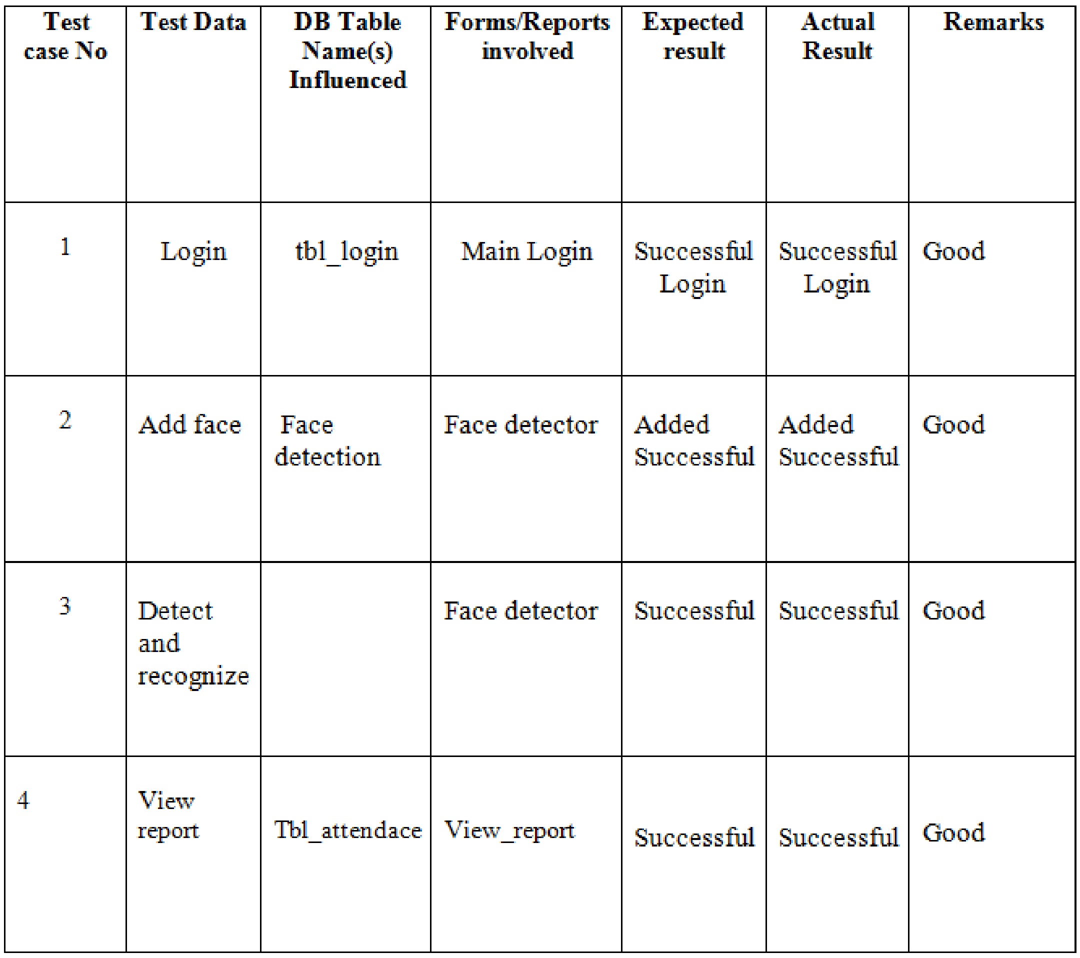
7.7 User Acceptance Testing

The system is validated by negotiating the existing and proposed system. This test evaluates the system in the real time environment with live data and finds it to be satisfied. This is done by the user. The various possibilities of the data are entered and response from the system is tested once the acceptance testing is signed off by the user.

7.8 Test Cases

A Test Case is a script, program, or other mechanism that exercises a software component to ascertain that a specific correctness assertion is true. In general, it creates a specified initial state, invokes the tested component in a specified way, observes its behavior, and checks to ensure that the behavior was correct.They are mainly of two types.

1. Formal test cases
2. Informal test cases



Test case No	Test Data	DB Table Name(s) Influenced	Forms/Reports involved	Expected result	Actual Result	Remarks
1	Login	tbl_login	Main Login	Successful Login	Successful Login	Good
2	Add face	Face detection	Face detector	Added Successful	Added Successful	Good
3	Detect and recognize		Face detector	Successful	Successful	Good
4	View report	Tbl_attendace	View_report	Successful	Successful	Good

Figure 3: Test case

7.9 Test Results

Each and every computer project start with a statement of the business needs and is then developed in progressively greater level of details. It is the purpose of testing to ensure that the communication between each level is verified and that the end project satisfies the business needs.

Testing a system requires more effort while developing it because it is one of the final steps. Early planning for this stage can ensure smooth and easy testing. Adequate preparation needs to be made before testing begins, so that it can be performed effectively.

The aim of testing is to prove that the developed system addresses the predefined processing requirements and will perform reliably and efficiently when running live. In the customer relationship management, testing is performed by providing test data to check the working of the system as specified. The complete system is tested to the satisfaction of users

7.10 Manual Testing

Manual testing is the process of manually testing software for defects. It requires a tester to play the role of an end user whereby they use most of the application's features to ensure correct behavior. To guarantee completeness of testing, the tester often follows a written test plan that leads them through a set of important test cases.

A key step in the process is, testing the software for correct behavior prior to release to end users. For small scale engineering efforts (including prototypes), exploratory testing may be sufficient. With this informal approach, the tester does not follow any rigorous testing procedure, but rather explores the user interface of the application using as many of its features as possible, using information gained in prior tests to intuitively derive additional tests. The success of exploratory manual testing relies heavily on the domain expertise of the tester, because a lack of knowledge will lead to incompleteness in testing. One of the key advantages of an informal approach is to gain an intuitive insight to how it feels to use the application.

A rigorous test case based approach is often traditional for large software engineering projects that follow a Waterfall model.However, at least one recent study did not show a dramatic difference in defect detection efficiency between exploratory testing and test case based testing.

Testing can be through black-, white- or grey-box testing. In white-box testing the tester is concerned with the execution of the statements through the source code. In black-box testing the software is run to check for the defects and is less concerned with how the processing of the input is done. Black-box testers do not have access to the source code. Grey-box testing is concerned with running the software while having an understanding of the source code and algorithms.

Static and dynamic testing approach may also be used. Dynamic testing involves running the software. Static testing includes verifying requirements, syntax of code and any other activities that do not include actually running the code of the program. Testing can be further divided into functional and non-functional testing. In functional testing the tester would check the calculations, any link on the page, or any other field which on given input, output may be expected. Non-functional testing includes testing performance, compatibility and fitness of the system under test, its security and usability among other things.

There are several stages.They are:

1. Unit Testing
2. Integration Testing
3. System Testing
4. User Acceptance Testing
5. Deployment Testing

8 SYSTEM SECURITY

Any computer-based system that manages sensitive information or causes actions that can improperly harm individuals is a target for improper or illegal penetration. Penetration spans a broad range of activities : hackers who attempt to penetrate systems for sport; disgruntled employees who attempt to penetrate for revenge; dishonest individuals who attempt to penetrate for illicit personal gain.

The proposed system, provide security technology attempts to verify that protection mechanisms built into a system will, in fact, protect it from improper penetration. The systems security must, of course, be tested for invulnerability from frontal attack, but must also be tested for invulnerability from rear attack.

The security and integrity of data is ensured in proposed system. The access to data is restricted to those who register with the system. While registering with the system, the users are provided with a username and password. Only the registered person can login into to the system. The username and password determine which user is login.

9 SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementations, design of the methods to achieve the changeover methods .Apart from planning major tasks of preparing the implementation are education and training of users. The more complex system is being implemented, the more involved will be the system analysis and design effort required just for implementation.

An implementation co-ordination committee based on politics of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. Implementation is the final and important phase. The system can be implemented only after through testing is done and it is found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system. The implementation plan includes a description of all activities that must occur to implement the system and to put it into operation .It indicates the personal responsible for the activities and prepares a time chart for implementing the system. The implementation plan consists of the following steps:

1. List all files required for implementation.
2. Identify all data required to build new files during the implementation.
3. List all new documents and procedures that go into the new system.

The implemented system has the following features:

1. Reduce data redundancy.
2. Ease of use
3. Controlled flow
4. Simplifies the management activities.

A critical phase in the system life cycle the successful implementation of the new system design. Implementation include all those activities that take place to convert the old system to new one. It is primarily concerned with user training, site preparation and fail conversion. The new system may be completely new, replacing and existing manual or automated system or it may be major modification to existing. In either case, proper implementation becomes necessary so that a reliable system based on the requirements of the organization can be provided. Successful implementation may not guarantee improvements in the organization using the new system, but improper installation can be prevented. It has been observed that even the best system cannot show good result if the analysts managing the implementation do not attended to every important detail. The only training required to operate this system is to develop a familiarity with the various features of the system and how to use it. The user where trained to enter the correct data in the correct places. No special training, other than this is required to operate the system. A person with little computer knowledge can operate the system. This is an area where the system analysts need to work with almost care.

10 SYSTEM MAINTENANCE

Once the software is delivered and deployed, the maintenance phase starts. Software requires maintenance because there are some residual errors remaining in the system that must be removed as they discovered. Maintenance involves understanding the existing software(code and related documents) , understanding the effect of change, making the changes, testing the new changes and retesting the old parts that were not changed. The complexity of the maintenance part makes maintenance the most costly activity in the life of software product. It is believed that almost all software that is developed has residual errors, or bugs in them. These errors need to be removed when discovered that leads to the software change. This is called corrective maintenance.Corrective maintenance measure pairing, processing of performance failures or making alterations because of previously ill-defined problems. Software undergoes change frequently even without bugs because the software must be upgraded and enhanced to include more features and provide more services. This also requires modification of the software. The changed software changes the environment, which in turn requires further change. This phenomenon is called Low of Software Evaluation.The keys to reduce the needs for maintenance are:

1. More accurately defining the users requirement during system development.
2. Preparation of system documentation in a better way.
3. Using more effective ways for designing processing logic and communicating it to project team members.
4. Making better use of existing tools and techniques.
5. Managing the system engineering process effectively.

During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called corrective maintenance.As the software is used recommendations for new capabilities, modifications to existing functions,and general enhancements are received from users.

10.0.1 Types of Software Maintenance

1. Corrective :

   Corrective maintenence of a software products become necessery to rectify the bugs while the system in use.

2. Adaptive:

   A software product might need maintenance when the customers need the product to run on new platforms,on new operating systeme,or when they need the product to be interfaced with new hardware or software.

3. Perfective:

   A software product needs maintenance to support the new features that users want it to support, to change different functionalities of the system according to the customers need ,or to enhance the performance of the system.

11 SYSTEM EVALUATION

Although system evaluation is an ongoing process throughout the performance testing effort, it offers greater value when conducted early in the test project. The intent of system evaluation is to collect information about the project as a whole, the functions of the system, the expected user activities, the system architecture, and any other details that are helpful in guiding performance testing to achieve the specific needs of the project.

1. Your need to evaluate and select software that meets your business requirements.

2. Your need to evaluate and select a partner that is capable of delivering the most benefit to your business from your software investment, as well as managing the risks inherent in system implementation projects.

3. Your time and ours is valuable; at each step along the way we will each decide whether or not it is beneficial to proceed.

To help you with your selection, this evaluation process is designed to give us both a clear understanding of the systems to be implemented and the corresponding benefits of the partnership. This information provides a foundation for collecting the performance goals and requirements, characterizing the workload, creating performance-testing strategies and classifieds, and assessing project and system risks. A thorough understanding of the system under test is critical to a successful performance-testing effort. The measurements gathered during later stages are only as accurate as the models that are developed and validated in this stage. The evaluation provides a foundation for determining acceptable performance; specifying performance requirements of the software, system, or component(s); and identifying any risks to the effort before testing even begins. System evaluation providing in these project is needed to evaluate and select the requirements and managing the risk in system implementation on project. Also it is valuable in time so that way it is beneficial in each steps.

12 CONCLUSION

The project entitled face detection attendence system was tested with proper date. The system is more helpful and has advantages over the existing system. The entire system is menu assisted and highly interactive. In this system, neat formatted reports can be printed within a short period of time. The system is very user friendly and reports are screen oriented. Accurate updating, data validation and integrity are observed in the system. The system was developed to overcome the difficulties encountered in presently used system. The development of this project underwent the various states of project developments like System analysis, System design, System testing and System implementation. After considering the various feasible solutions, the most feasible one was selected for designing taking into consideration the time and effciency constraints..

12.1 SCOPE FOR FUTURE ENHANCEMENTS

In future we can expect the modified version of Face detection attendence system. The system is very flexible for further up gradation with additional requirement of the company, the Asp.net and MySQL makes this modifications very easily It is also possible to involve more functions into the system. This flexibility makes this system widening its scope. All day to day work can be done with much more ease and efficiency.

Throughout all the phase changes keep cropping up and the system must be able to adjust itself to the changing situation. The system has been developed keeping this in mind. The system has been so good software must be able to incorporate future modifications and enhancement. Developed but changes in configuration leaves minimum impact on the performance.

The application developed can be done with ease. The system has the capability for easy integration with other system. New modules can be added to the existing system with less effort.

The database and the information can be updated to the latest coming versions. There are also possibilities for enhancing and further developing the project with the latest information .The database and the information can be updated to the latest coming versions. There are also possibilities for enhancing and further developing the project with the latest information and needs of the portal.

13 APPENDIX

13.1 APPENDIX A

13.1.1 Sample Source Code / Pseudo Code

```
1. LOGIN
.using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Web;
usingSystem.Web.UI;
usingSystem.Web.UI.WebControls;
usingSystem.Data.SqlClient;
usingSystem.Data;
usingSystem.Configuration;
publicpartialclassLogin_new : System.Web.UI.Page
{
protectedvoidPage_Load(object sender, EventArgs e)
{
if (!IsPostBack)
{
Session["E1"] = "";
Session["M1"] = "";
Session["R1"] = "";
Session["A1"] = "";
//Session["A2"] = "";
//Session["B2"] = "";
Session["D1"] = "";
}
}
protectedvoid Button1_Click(object sender, EventArgs e)
{
SqlConnection con = newSqlConnection(ConfigurationManager.
ConnectionStrings["ConnectionString"].ConnectionString );
con.Open();
SqlCommandcmd = newSqlCommand("select * from
Tbl_Login where UserName='" + TextBox1.Text +"' and
Password='"+ TextBox2.Text +"'",con);
SqlDataAdaptersda = newSqlDataAdapter(cmd);
DataTabledt = newDataTable();
sda.Fill(dt);
con.Close();
if (dt.Rows.Count == 0)
{
Response.Write("<script>alert('Invalid UserName or Password');
</script>");
}
else
{
Session["uid"] = dt.Rows[0][0].ToString();
Response.Redirect("Entity_new.aspx");
}
}
```

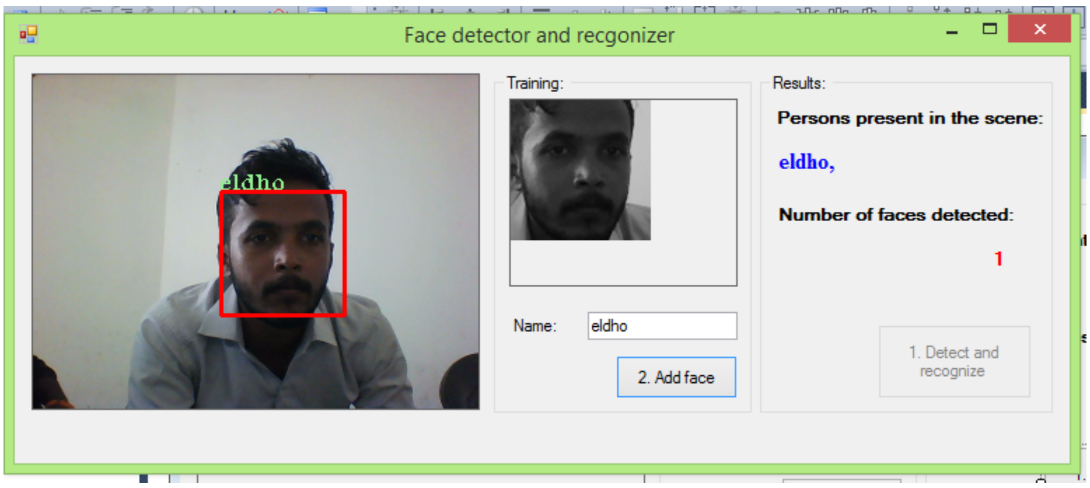
```
 2.FACE RECGANITION
using System;
using System.Diagnostics;
using Emgu.CV.Structure;

namespace Emgu.CV
{
   /// <summary>
   /// An object recognizer using PCA (Principle Components Analysis)
   /// </summary>
   [Serializable]
   public class EigenObjectRecognizer
   {
      private Image<Gray, Single>[] _eigenImages;
      private Image<Gray, Single> _avgImage;
      private Matrix<float>[] _eigenValues;
      private string[] _labels;
      private double _eigenDistanceThreshold;

      /// <summary>
      /// Get the eigen vectors that form the eigen space
      /// </summary>
      /// <remarks>The set method is primary used for deserialization,
 do not attemps to set it unless you know what you are doing</remarks>
      public Image<Gray, Single>[] EigenImages
      {
         get { return _eigenImages; }
         set { _eigenImages = value; }
      }

      /// <summary>
      /// Get or set the labels for the corresponding training image
      /// </summary>
      public String[] Labels
      {
         get { return _labels; }
         set { _labels = value; }
      }

      /// <summary>
      /// Get or set the eigen distance threshold.

      /// </summary>
      public double EigenDistanceThreshold
      {
         get { return _eigenDistanceThreshold; }
         set { _eigenDistanceThreshold = value; }
      }

      /// <summary>
      /// Get the average Image.
      /// </summary>

      public Image<Gray, Single> AverageImage
      {
         get { return _avgImage; }
         set { _avgImage = value; }
      }

      /// <summary>
      /// Get the eigen values of each of the training image
      /// </summary>

      public Matrix<float>[] EigenValues
```

```
      {
         get { return _eigenValues; }
         set { _eigenValues = value; }
      }

      private EigenObjectRecognizer()
      {
      }


      /// <summary>

      public EigenObjectRecognizer(Image<Gray, Byte>
[] images, ref MCvTermCriteria termCrit)
         : this(images, GenerateLabels(images.Length), ref termCrit)
      {
      }

      private static String[] GenerateLabels(int size)
      {
         String[] labels = new string[size];
         for (int i = 0; i < size; i++)
            labels[i] = i.ToString();
         return labels;
      }

      /// <summary>
      /// Create an object recognizer using the specific tranning
 data and parameter
, it will always return the most similar object
      /// </summary>

      /// <param name="labels">

         : this(images, labels, 0, ref termCrit)
      {
      }
      public EigenObjectRecognizer(Image<Gray, Byte>[] i
mages, String[] labels, double eigenDistanceThreshold
, ref MCvTermCriteria termCrit)
      {
         Debug.Assert(images.Length == labels.Length,
"The number of images should equals the number of labels");
         Debug.Assert(eigenDistanceThreshold >= 0.0, "
Eigen-distance threshold should always >= 0.0");

         CalcEigenObjects(images, ref termCrit,
 out _eigenImages, out _avgImage);

         /*
         _avgImage.SerializationCompressionRatio = 9;

         foreach (Image<Gray, Single> img in _eigenImages)
             //Set the compression ration to best compression.
The serialized object can therefore save spaces
             img.SerializationCompressionRatio = 9;
         */

         _eigenValues = Array.ConvertAll<Image<Gray, Byte>,
 Matrix<float>>(images,
             delegate(Image<Gray, Byte> img)
             {
                return new Matrix<float>
(EigenDecomposite(img, _eigenImages, _avgImage));
             });

         _labels = labels;

         _eigenDistanceThreshold = eigenDistanceThreshold;
      }

      \#region static methods

 ref MCvTermCriteria termCrit, out Imag
e<Gray, Single>[] eigenImages, out Image<Gray, Single> avg)
      {
         int width = trainingImages[0].Width;
         int height = trainingImages[0].Height;

         IntPtr[] inObjs = Array.ConvertAll
<Image<Gray, Byte>, IntPtr
>(trainingImages, delegate(Image<Gray, Byte> img)
 { return img.Ptr; });

         if (termCrit.max_iter <= 0 || termCrit.max_iter
> trainingImages.Length)
            termCrit.max_iter = trainingImages.Length;

         int maxEigenObjs = termCrit.max_iter;
         \#region initialize eigen i eigenImages =
 new Image<Gray, float>[maxEigenObjs];
     for (int i = 0; i < eigenImages.Length; i++)
      eigenImages[i] = new Image<Gray, float>(width, height);
       IntPtr[] eigObjs = Array.ConvertAll<Image<Gray, Single>, IntPtr>
(eigenImages, delegate(Image<Gray, Single> img) { return img.Ptr; });
         \#endregion

       avg = new Image<Gray, Single>(width, height);

       CvInvoke.cvCalcEigenObjects(
             inObjs,
           ref termCrit,
         eigObjs,
           null,
             avg.Ptr);
      }

      {
         return CvInvoke.cvEigenDecomposite(
             src.Ptr,

             Array.ConvertAll<Image<Gray, Single>, IntPtr
>(eigenImages, delegate(Image<Gray, Single> img) { return img.Ptr; }),
             avg.Ptr);
      }
      \#endregion


      public Image<Gray, Byte> EigenProjection(float[] eigenValue)
      {
         Image<Gray, Byte> res = new Image<Gray, byte>
(_avgImage.Width, _avgImage.Height);
         CvInvoke.cvEigenProjection(
             Array.ConvertAll<Image<Gray, Single>, IntPtr>
(_eigenImages, delegate(Image<Gray, Single> img) { return img.Ptr; }),
             eigenValue,
             _avgImage.Ptr,
             res.Ptr);
         return res;
      }


      {
         using (Matrix<float> eigenValue =
 new Matrix<float>(EigenDecomposite(image, _eigenImages, _avgImage)))
            return Array.ConvertAll<Matrix<float>, float>(_eigenValues,
                delegate(Matrix<float> eigenValueI)
                {
                   return (float)CvInvoke.cvNorm
(eigenValue.Ptr, eigenValueI.Ptr
, Emgu.CV.CvEnum.NORM_TYPE.CV_L2, IntPtr.Zero);
                });
      }


      {
         float[] dist = GetEigenDistances(image);

         index = 0;
         eigenDistance = dist[0];
         for (int i = 1; i < dist.Length; i++)
         {
            if (dist[i] < eigenDistance)
            {
               index = i;
               eigenDistance = dist[i];
            }
         }
         label = Labels[index];
      }

      /// <summary>
      /// Try to recognize the image and return its label
      /// </summary>
      /// <param name="image">The image to be recognized</param>
      /// <returns>
      /// String.Empty, if not recognized;
      /// Label of the corresponding image, otherwise
      /// </returns>
      public String Recognize(Image<Gray, Byte> image)
      {
       int index;
         float eigenDistance;
         String label;
         FindMostSimilarObject(image, out index,
out eigenDistance, out label);

   return (_eigenDistanceThreshold <= 0
|| eigenDistance < _eigenDistanceThreshold )
  ? _labels[index] : String.Empty;
      }
   }
}
```

13.2 APPENDIX B

13.2.1 SCREEN SHOTS

1. Home Page

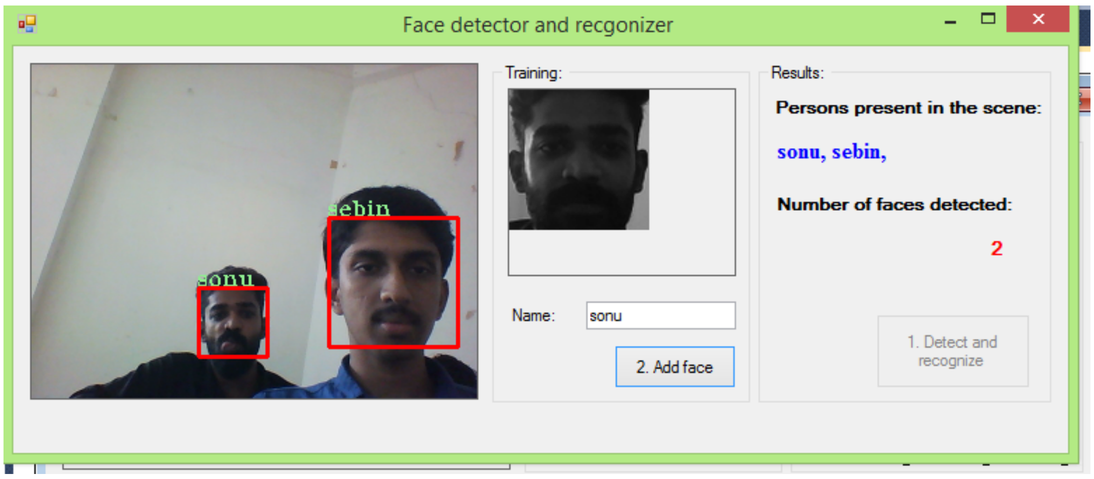


Persons present in the scene:

eldho,

Number of faces detected:

1



Persons present in the scene:

sonu, sebin,

Number of faces detected:

2

13.3 APPENDIX C

13.3.1 Acronyms

- PHP: HyperText Preprocessor
- GUI: Graphical User Interface
- MVC: Model View Controller
- IDE: Integrated Development Environment
- RAM: Random Access Memory
- SDK: Software Development kit
- CSS:Cascading Style Sheet
- SQL:Structured Query Language
- PCA :Principal Component Analysis

13.3.2 Bibliography

1. System Analysis and Design - ELIAS M.AWAD
2. Software Engineering -ROGER.S.PRESSMAN
3. http://www.sourcecodester.com
4. http://www.w3schools.com
5. http://www.google.co.in
6. http://www.wikipedia.org